SPECIFICATION

Please amend the Specification as follows:

Please replace the paragraph beginning on Page 3, line 17 with the following:

A system for cache coherency comprises a memory. The memory comprises a plurality of data items and a plurality of directory information items, each data item uniquely associated with one of the plurality of directory information items. Each of the plurality of data items is configured in accordance with one of a plurality of access modes. Each of the plurality of directory information items comprises indicia of the access mode of its associated data item. A multiplexer couples to the memory and comprises a multiplex ratio. A plurality of buffers couple to the multiplexer and to the memory. The multiplex ratio is a function of the number of buffers in the plurality of buffers. A plurality of multiplexer/demultiplexers (MDMs) each uniquely couple to a different one of the plurality of buffers. A plurality of processing elements couple to the memory; each of the processing elements uniquely couples in a point-to-point connection to a different one of the plurality of MDMs. Each of the processing elements is configured to transmit a data request to its associated MDM, the data request identifying one of the plurality of data items and an access mode. The memory is configured to transmit a data response to each of the processing elements in response to a data request, the data response comprising the identified data item and its associated directory information. Each of the processing elements is further configured to receive the data response and to compare the associated directory information with the access mode of the data request and in the event that the associated directory information and the access mode of the data request are not compatible, to initiate coherence actions for the requested data item.

Please replace the paragraph beginning on Page 4, line 6 with the following:

FIGURE 1 illustrates an exemplary diagram of a high bandwidth SMP (symmetrical processor);

Please replace the paragraph beginning on Page 4, line 8 with the following:

FIGURE 2 illustrates an exemplary custom dynamic random access memory (DRAM) with a 6:1 multiplexer and associated buffers;

Please replace the paragraph beginning on Page 5, line 16 with the following:

FIGURE 13 illustrates an exemplary block diagram for ~~certain~~ a page cache method function within an extended method for accessing custom DRAM; and

Please replace the paragraph beginning on Page 6, line 32 with the following:

MPP is similar to symmetric processing (SMP). One main difference is that in SMP systems, all the CPUs share the same memory, whereas in MPP systems, each CPU has its own memory. MPP systems are therefore more difficult to program because the application must be divided in such a way that all the executing segments can communicate with each other. On the other hand, MPP systems do not suffer from the bottleneck problems inherent in SMP systems when all the CPUs attempt to access the same memory at once. MPP[[']]s retain the programming ease of SMP[[']]s or [[UMA]] NUMA type systems, in that all CPUs in the system are subjected to the same latency and bandwidth restrictions with respect to accessing the system's memory and I/O channels. The CDRAM can be treated as a closer main memory to the processor elements (PEs) or as a level of the cache hierarchy.

Please replace the paragraph beginning on Page 9, line 10 with the following:

When a node requests a memory unit by a request over the point-to-point link, the memory unit is returned with the additional information. ~~Requested~~ The requested memory unit can be a superset (or subset) of the granularity in which memory units and information associated with them [[is]] are stored in memory system. In addition, the information associated with the request is updated, for example, to indicate that a specific memory unit, for example, ~~indicating that the memory unit~~ has been requested by a processor, possibly further indicating the mode, for example, in shared mode, and/or possibly indicating which processor (or node) has performed the request.

Please replace the paragraph beginning on Page 9, line 22 with the following:

When the requested memory unit is returned to the requesting node (processor), the processor uses logic circuitry (or program controlled means), for example, in a memory flow controller[[or]], DMA controller, or CDRAM access controller, to test whether the information about the memory unit accessed is consistent with the desired/requested access mode.

Please replace the paragraph beginning on Page 10, line 3 with the following:

The resolution action will perform coherence actions, for example, via a slower operation bus [[which]] that connects one or more nodes, or by performing coherence action [[thru]] through a special message-passing protocol (for example, by raising a single protocol service request line, and depositing the requested coherence action in a known location, for example, in a specially designated storage facility[[.]]), and preferably updates the information in system memory to indicate the new state after the coherence action has been performed.

Please replace the paragraph beginning on Page 10, line 12 with the following:

In one aspect of this invention, the data size of transferred memory units exceeds the basic storage unit for coherence actions. When a node fetches a memory unit, it is cached in at least one page cache, together with all meta information. In the following description, we presume that the transfer size is a page: [[When]] when a request can be satisfied from at [[lest]] least one local page cache, then no request to system memory is made.

Please replace the paragraph beginning on Page 10, line 20 with the following:

To ensure correct protocol processing, additional protocol actions are possible. In one possible embodiment, the system memory stores the addresses of cached pages and appends them to each memory request. Cached pages may then be considered in shared (or exclusive) mode in particular variants of the protocol. When a page is evicted from the page cache, a directory information update is performed on blocks ~~contained~~ containing the page in system memory, to indicate all additional requests, which were made satisfied from the page cache, and not propagated to main memory. In addition, the address indicating the contents of a node's (processors) page cache is reset.

Please replace the paragraph beginning on Page 11, line 26 with the following:

Processing element PE 130 contains memory interface controller 134, connected to the buffer BUF 136 by a bus BUS 132, such that the BUF 136 is linked to custom DRAM device CDRAM_105. Processing element PE 140 contains memory interface controller 144, connected to the buffer BUF 146 by a bus.

Please replace the paragraph beginning on Page 11, line 32 with the following:

The BUS 142 is configured in a manner such that the BUF 146 is linked to custom DRAM device CDRAM 105, and, processing element PE 150 contains memory interface controller 154, connected to the buffer BUF 156 by a bus BUS 152, such that the BUF_156 is linked to custom DRAM device CDRAM 105.

Please replace the paragraph beginning on Page 12, line 19 with the following:

Turning now to FIGURE 2, disclosed is an exemplary system diagram showing the use of an exemplary custom 64-megabyte DRAM with a 6:1 multiplexer and associated buffers in one exemplary system, and an exemplary interconnection detail.

Please replace the paragraph beginning on Page 12, line 24 with the following:

This exemplary DRAM 200 is connected to four processing elements, PE 210, PE 230, PE 250, and PE 270. Each of the four PEs is connected to an adjacent multiplexer/demultiplexer by a point-to-point link. All links in this example are bi-directional links, multiplexing and de-multiplexing (that is,[[,]] joining and separating) data signals according to a programming scheme.

Please replace the paragraph beginning on Page X, line X with the following:

PE 210 is attached to MUX 215 via BUS 212. ~~PE2 30~~ PE 230 is attached to MUX 235 via BUS 232. PE 250 is attached to MUX 255 via BUS 252 and PE 270 is attached to MUX 275 via BUS 272. Similarly, in one exemplary embodiment, external memory is optionally connected to the system via an I/O processor (not shown, and not limited to a single main memory or single I/O processor), via BUS 207 to MUX 205. Likewise, in one exemplary embodiment, BUS 283

optionally conducts signals between MUX 285 and a visualizer or other I/O devices (not shown for clarity). (A visualizer is the interface to a graphics display device.)[[.]]

Please replace the paragraph beginning on Page 14, line 4 with the following:

Turning to FIGURE 3, disclosed is an exemplary processing element PE, consisting of a memory interface controller (DMA 328), at least one Processing Unit (PU 300A) and optionally at least one Auxiliary Processing Unit (APU 300B). FIGURE 3 additionally provides interconnection detail connecting the memory interface controller to PU and APU units. The PU consists of a typical set of function blocks, optionally including, but not limited to, an Instruction Translation Look-aside Buffer (ITLB 302), Instruction Cache (I$ 308), Control Logic (CTRL 306), General Purpose Register File (GPR 310), Fixed Point Unit (FXU 312), Single Instruction-Multiple Data (SIMD) processing unit (SIMD 314), SIMD Vector Register File (SRF 316), Data Translation Look-aside Buffer (DTLB 320), Data Cache D$ 324, and, optionally, a page cache P$ 326; and within the APU 300B, control logic CTRL 332, [[,]] SIMD processing unit 338, SIMD vector register file 340, Data Local Store (LSD 342), Instruction Local Store (LSI 334). In one alternative embodiment, the memory management and translation functionality (such as ITLB 302 and DTLB 320) are contained in memory interface controller 328. (In one embodiment, this memory interface controller is referred to by the name DMA controller. In yet another embodiment, this memory interface controller is referred to by the name memory flow controller.)

Please replace the paragraph beginning on Page 14, line 29 with the following:

When a line is evicted from the cache hierarchies in the PE 210 through PE 270 of FIGURE 2, the coherence directory can either be updated immediately (at additional transaction cost when

the line is freed), or updated in a lazy manner, when a remote PE requests a line which is no longer present (at a transaction cost penalty for the requesting PE when the line is requested).

Please replace the paragraph beginning on Page 15, line 16 with the following:

For example, a penalty can occur when P$ 326 sub-lines are requested from remote PEs, but are never used by the requesting PE. This can occur due to false sharing on a CDRAM line basis, and given the coarse nature of the CDRAM line may be common for many applications that have not been optimized for this environment. Updates to the coherence directory may be frequent due to request and release of successive memory units. The first issue can be solved by providing the P$ 326 with those sub-lines that are available and need not be requested using coherence actions. Thus, prefetch is only performed on available lines, reducing the cost of false sharing. In an optimized embodiment, predictors might be used to determine if certain sub-lines, which are not available, should be requested, while identifying other sub-lines, which are not subject to requests, by the P$ 326 file mechanism.

Please replace the paragraph beginning on Page 16, line 29 with the following:

CD 405 also receives/sends indicia through a bus PB 410 at the same inferred target rate, to external I/O (BIF) ports. Resident and transient data is processed independently by PEs, PE 422, PE 432, PE 442 and PE 452, each with its own bi-directional CDRAM to Chip Stacked Ports ([[CBIF]]CPIF), CPIF 420, CPIF 430, CPIF 440 and CPIF 450, all operating at an inferred rate of ¼ terabyte [that is, 250 gigabytes] per second (A terabyte is a measure of computer storage capacity and is 2 to the 40th power or approximately a thousand billion bytes - that is, a thousand gigabytes).

Each PE device is of approximately 70mm2 dimensions and uses 0.10 micron SOI [silicon-on-insulator] Process Technology (CMOS 10S) [a proprietary technology developed by IBM].

Please replace the paragraph beginning on Page 22, line 9 with the following:

In one embodiment, this includes using a (possibly slower) coherence bus provided[[in a]]. In another preferred embodiment, this involves operations performed in a dedicated state machine, microcode, millicode, firmware, or software. In one embodiment, the coherence bus is a physical multi-drop coherence bus, in a further embodiment, the coherence bus is a logical coherence bus composed of multiple point-to-point links. In yet another embodiment there is provided a coherence ring. In other embodiments, coherence actions use special signaling primitives between nodes provided by the system (such as, including but not limited to, processor to processor interrupts), or a messaging and signaling system provided in the memory. Eventually, coherence actions obtain the data with the required access mode and provide them to at least one processor core in the node for either processing or storage with the requested access mode.

Please replace the paragraph beginning on Page 23, line 24 with the following:

In step 860, there is performed a third test, to see if the data can be successfully acquired in exclusive mode, that is,[[,]] the directory information does not indicate any use in shared or exclusive mode. If so, step 862 indicates success. This is possible because acquiring data items in exclusive mode is compatible with memory-resident state only, but not when data are maintained in any state in another node.

Please replace the paragraph beginning on Page 24, line 6 with the following:

In a first step 870, the shared and exclusive conditions are determined. (Specific bits in this format are used to perform efficient coherence actions, but are not necessary for the execution of step 820.) In step 872, a first test is performed to determine if the directory information has a legal form. If not, control transfers to step 874 which invokes special condition handling mechanisms implemented in conjunction with FIGURE 8, such as error handling, and protocol retry. In step 876, a second test is performed to test if the data items are required in shared mode, and are not in exclusive mode. If so, success is indicated in step 878. In step 880, there is performed a third test, to see if the data can be successfully acquired in exclusive mode, that is,[[,]] the directory information does not indicate any use in shared or exclusive mode. If so, step 882 indicates success. Otherwise, coherence actions are necessary, and this is indicated in step 884. The statements associated with these calls are as follows.

Please replace the paragraph beginning on Page 25, line 13 with the following:

wherein the subscription operator [I] indicates the bit numbered I of a bit vector, the operator = indicates assignment, and the operator OR corresponds to the logical OR of two Boolean values. The variables request.exclusive and request.shared indicate whether the request was for shared or exclusive access mode. [[(]]In another embodiment, the request is indicated by a single bit.[[).]]

Please replace the paragraph beginning on Page 26, line 11 with the following:

wherein the subscription operator [I] indicates the bit numbered I of a bit vector, and the operator = indicates assignment. The variables request.exclusive and request.shared indicate whether the request was for shared or exclusive access mode, and request.node indicates the number of the requesting node. [[(]]In another embodiment, the request is indicated by a single bit.[[)]]

Please replace the paragraph beginning on Page 27, line 9 with the following:

There is now set forth another exemplary embodiment of the preferred embodiment, according to the instruction set of the preferred embodiment. According to the exemplary embodiment, there are [[at]] four nodes connected to a shared memory hierarchy level via point-to-point links, the links providing means for requesting data for immediate processing or local storage in a cache for future processing. Each node consists of at least one processor. In one embodiment, each node is a heterogeneous system, consisting of at least one processing unit and one auxiliary processing unit. In another embodiment, there is provided a heterogeneous SMP, wherein at least one node has at least one first processing unit, and at least one node has at least one second processing unit (such as an auxiliary processing unit). According to this embodiment, there is supported prefetching capability. In one embodiment, a page containing a data item is prefetched. In another embodiment, other sequences of data items are prefetched. For making the features of this invention apparent, we will describe the embodiments in the context of prefetching a page surrounding the specific data item request.

Please replace the paragraph beginning on Page 30, line 11 with the following:

In step 1230, when it is determined that the requested data has been successfully retrieved from shared memory hierarchy level 1000 with permissions [[which]] <u>that</u> are compatible with the requested access mode, [[and]] the data is provided in step 1230 to at least one processor core in the node for either processing or storage with the requested access mode. In step 1235, the received page and associated directory information is optionally stored in a prefetch page cache.

Please replace the paragraph beginning on Page 32, line 7 with the following:

Referring now to step 1225, a method in accordance with FIGURES 8A and 8B can be used, employing a directory for the specific memory subunit being requested. In another embodiment, the methods of FIGURES 8A and 8B are extended to include prefetch information in the testing steps. In one embodiment, this is achieved by extending steps 850 and 870, respectively, to incorporate the prefetch address information into the tests being performed on state variables "shared" and "exclusive". In another embodiment, such steps incorporating prefetch information in directory information is performed by logic in the shared memory hierarchy level 1000.

Please replace the paragraph beginning on Page 34, line 23 with the following:

Turning now to FIGURE 13, there are shown exemplary methods implemented by the shared memory hierarchy level 1000. In a first method 1305, the shared memory hierarchy level 600 receives a request in a first step 1310,[[,]] for example, in accordance with the exemplary FIGURE 11, over a point to point link from a node. In a second step 1315, a page containing the requested data item as well as other data items hierarchically composing a page selected from data items 1005-1012 are returned to the requesting node, in conjunction with the directory information state 1015-1022 associated with those data items being transferred, as well as contents of prefetch address registers. In one embodiment, all prefetch registers are transmitted. In another embodiment, only those registers having a possible conflict are indicated. In another embodiment, the contents of the prefetch registers 1030-1036 are combined with the directory information 1015-1022 before being transferred in step 1315. In another step [[920]] <u>1320</u>, directory information is updated. In another step 1325, page cache information such as prefetch address registers 1030-1036 are updated. Steps 1315, 1320 and 1325 are preferably implemented atomically with respect to other

protocol transactions, in particular to other execution instances of steps 1315-1320, or 1340-1345 on behalf of other nodes.

Please replace the paragraph beginning on Page 35, line 22 with the following:

It will also be apparent that according to the embodiments set forth in this invention, the memory is used as a central integration point. In one aspect of integration, coherence is performed not by a central directory controller, or through a snoop of a central bus, but by reading [[and]] an updated directory information of a memory component in the system.

Please replace the paragraph beginning on Page 36, line 5 with the following:

In another aspect of this invention, chip stacking is used to physically integrate the system, processing element chips being physically stacked onto or with a memory component, and the memory component serves as physical integration point. In one specific embodiment, C4 connectors are used to connect chip-stacked processing elements and memory component chips. In another aspect of this invention, wire-bonding is used to connect chip-stacked processing elements and memory component chips.

Please delete the paragraph beginning on Page 37, line 6, which begins with, "It is understood that the present invention can take many forms and implementations. . ."

Please delete the paragraph beginning on Page 37, line 16, which begins with, "Having thus described the present invention by reference to certain of its salient characteristics. . ."

Please replace the paragraph beginning on Page 44, line 6 (The Abstract) with the following:

A system for cache coherency comprises a memory. The memory comprises a plurality of data items and a plurality of directory information items, each data item uniquely associated with one of the plurality of directory information items. Each of the plurality of data items is configured in accordance with one of a plurality of access modes. Each of the plurality of directory information items comprises indicia of the access mode of its associated data item. A multiplexer couples to the memory and comprises a multiplex ratio. A plurality of buffers couple to the multiplexer and to the memory. The multiplex ratio is a function of the number of buffers in the plurality of buffers. A plurality of multiplexer/demultiplexers (MDMs) each uniquely couple to a different one of the plurality of buffers. A plurality of processing elements couple to the memory; each of the processing elements uniquely couples in a point-to-point connection to a different one of the plurality of MDMs. Each of the processing elements is configured to transmit a data request to its associated MDM, the data request identifying one of the plurality of data items and an access mode. The memory is configured to transmit a data response to each of the processing elements in response to a data request, the data response comprising the identified data item and its associated directory information. Each of the processing elements is further configured to receive the data response and to compare the associated directory information with the access mode of the data request and in the event that the associated directory information and the access mode of the data request are not compatible, to initiate coherence actions for the requested data item. A method for cache coherency is also provided.